CS SCHOLARS SEMINAR

March 7 and March 9, 2017

1. For each of the expressions in the table below, write the output displayed by the interactive Python interpreter when the expression is evaluated. The output may have multiple lines. If more than 3 lines are displayed, just write the first 3. If an error occurs, write Error. If evaulation would run forever, write Forever.

   The first two rows have been provided as examples.

   Assume that you have started `python3` and executed the following statements (which do not cause errors):

```python
class Pet: pass

class Dog(Pet):
    all_dogs = []
    def __init__(self, name):
        self.name = name
        self.all_dogs.append(self)

    def bark(self, name):
        times_barked = 0
        self = self
        def count(name):
            nonlocal times_barked
            print("Hi ", name, " my name is ", self.name)
            times_barked += 1
            return times_barked
        self.bark = count
        return self.bark(name)

    def __repr__(self):
        return "Count " + self.name

    def __str__(self):
        return "Queen " + self.name

class Puppy(Dog):
    def __init__(self, name):
        self.name = name
        self.all_dogs[0].all_dogs.append(self)
```

```python
    def bark(self, name):
        print("Hi " + name.name + " my name is ", self)

bob = Dog("bob")
steve = Puppy("steve")
print(bob.bark(steve))
# Hi  Queen steve  my name is  bob
print(steve.bark(bob))
# Hi bob my name is  Queen steve
steve.bark = bob.bark
print(steve.bark(bob))
# Hi  Queen bob  my name is  bob
def chase(self, toy):
    print("Yay! ", self, " caught ", toy)
    return self
Puppy.chase = chase
steve.chase("ball")
# Yay!  Queen steve  caught  ball
# Count steve

steve.chase = chase
# steve.chase()
# Error

# len(Pet.all_dogs)
# Error

print(Dog.bark(steve, bob))

print(steve.bark(bob))

steve.times_barked
```